공고특허10-0253242

(19)대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) Int. Cl. [7]
G06F 17/28

(45) 공고일자 2000년04월15일
(11) 공고번호 10-0253242
(24) 등록일자 2000년01월22일

(21) 출원번호	10-1997-0078899	(65) 공개번호	특1999-0058745
(22) 출원일자	1997년12월30일	(43) 공개일자	1999년07월15일

(73) 특허권자 엘지전자주식회사 구자홍
서울특별시 영등포구 여의도동 20번지

(72) 발명자 곽종근
서울특별시 서초구 서초1동 강남로얄 가동 지하 102호
강윤선
서울특별시 강남구 일원2동 대청아파트 302동 403호

(74) 대리인 박장원

심사관 : 이은철

(54) 프래그먼트 콤비네이션 방법

요약

본 발명은 프래그먼트 콤비네이션 방법에 관한 것으로 특히, 구문 분석시 입력 문장의 중심 의미를 벗어나지 않는 가능한 범위까지 부분 해석 트리를 조합하여 전체 문장의 중심 의미를 근사시키도록 함으로써 의미 추출의 정확성 및 효율성을 향상시키도록 창안한 것이다. 이러한 본 발명은 구문 분석이 실패한 경우 부분 분석 트리를 이용하여 경로를 형성하는 경로 형성 단계와, 상기에서 생성된 경로중 문장의 중심 의미를 추출할 수 있는 가능한 범위까지 경로를 선택하는 단계와, 상기에서 선택된 경로에 대해 문장의 의미를 추출하는 의미 추출 단계를 수행함을 특징으로 한다.

대표도

도1

명세서

도면의 간단한 설명

도 1은 본 발명의 실시를 위한 신호 흐름도.

도 2는 부분 분석 트리를 보인 예시도.

도 3은 도 2에서 리스트 구조를 어레이화한 표.

도 4는 도 2에서 경로 선택을 위한 신호 흐름도.

도 5는 도 4의 과정을 수정한 신호 흐름도.

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야 종래기술

본 발명은 자연어 처리 시스템에 관한 것으로 특히, 구문 분석 방법에 있어서 프래그먼트 콤비네이션 방법에 관한 것이다.

자연어 처리 기능이 포함된 시스템은 일반적으로 입력 문장의 의미 구조를 추출하여 그 의미에 따라 필요한 다른 작업을 수행한다.

따라서, 입력문의 의미를 추출하는 것은 자연어 처리 시스템의 기본적인 과제로 볼 수 있다.

입력문의 의미를 추출하기 위해서는 자연어 처리의 많은 과정을 거치게 되는데, 문장의 의미를 추출하려면 구문 분석 과정을 거쳐야 한다.

상기 구문 분석 과정은 입력문이 정해진 문법에 맞는가를 검사하는 과정으로, 입력문이 정의된 문법에 맞으면 구문 분석에 성공하고 그렇지 않으면 구문 분석은 실패하게 된다.

구문 분석에 실패하는 이유는 입력문이 실제로 잘못된 경우, 미등록어 또는 자연어 자체를 완전하게 기술하는 문법이 존재하지 않는 경우가 발생하기 때문이다.

즉, 자연어 처리 시스템(NLP ; Natural Language Processing System)에서 구문 분석이 실패할 가능성은 언제나 존재한다.

따라서, 구문 분석의 실패에 좀더 유연한 시스템을 설계할 필요성 때문에 구문 분석이 실패한 경우에 적절하게 대응하기 위한 연구가 진행되어 왔다.

구문 분석의 실패에 대응하기 위한 해결 방법은 자연어 처리 시스템이 적용되는 분야에 따라 달라지며, 이를 예를 들면 아래와 같다.

먼저, 문맥적 히스토리가 존재하는 시스템은 이에 근거하여 어느 정도의 기대가 가능함으로 잘못된 입력(ill-formed input)에 대한 처리를 위해서 더 많은 정보를 이용할 수 있다.

또한, 사용자와 대화(interaction)를 갖는 대화형 시스템은 잘못된 입력에 대해 시스템이 구문 분석을 못하는 상황을 사용자에게 설명함으로써 사용자가 정확한 입력문을 새로 제공하도록 유도하는 것으로 잘못된 입력문을 처리할 수 있는 방법이다.

한편, 구문 분석이 성공한 경우에는 그 결과로 전체 문장의 내부 구조를 표현한 구문 분석 트리를 결과로 얻을 수 있지만 만일, 구문 분석이 실패한 경우에는 실패한 그 순간부터 분석 가능하였던 부분들의 부분적인 분석 트리(Partial Parse Trees)만을 얻을 수 있다.

따라서, 구문 분석이 실패한 경우에 생기는 부분 분석 트리들을 이용하여 분석이 성공하는 경우 문장의 의미를 근사시키는 방법 즉, 구문 분석이 보톰업(bottom-up) 방식으로 진행되어 파싱의 중간 결과가 부분 해석 트리로 존재함으로 이를 조합하여 전체 문장의 의미를 근사시키는 방법이 제시되었다.

예로, 사용자의 메모 일정을 관리하는 NTA(Note Talking Agent) 시스템이 있다.

이러한 방법은 NTA 가 적용되는 범위의 특성상 문맥적 지식이 존재하지 않아 문맥적 지식의 관리가 필요없으므로 비교적 간단하면서 높은 효과를 얻을 수 있다.

그러나, 부분 분석 트리는 입력문의 전체 내용을 포함하는 것이 아니라 특정 구의 분석 트리를 나타내고 있기 때문에 이것들로부터 직접 전체 문장의 의미를 파악할 수 없다.

따라서, 구문 분석이 실패한 상황에서 전체 입력 문장을 커버(cover)하기 위한 프래그먼트(fragment)를 선택하여 구문 분석을 수행하는 방법으로, Jesen 의 EPISTLE 시스템이 제시되었다.

이러한 프래그먼트 콤비네이션(Fragment Combination) 방법은 구문 분석이 실패한 경우에도 휴리스틱(Heuristics)을 이용하여 분석에 성공한 경우와 유사한 결과를 얻을 수 있도록 한다.

발명이 이루고자하는 기술적 과제

그러나, 종래의 프래그먼트를 선택하는 기술은 하나로 통합된 의미를 추출하는 것이 아니므로 선택된 프래그먼트를 연결하는 작업을 하지 않아 정확한 의미 분석이 어려운 문제점이 있다.

따라서, 본 발명은 종래의 문제점을 해결하기 위하여 구문 분석시 입력 문장의 중심 의미를 벗어나지 않는 가능한 범위까지 부분 해석 트리를 조합하여 전체 문장의 중심 의미를 근사시키도록 함으로써 의미 추출의 정확성 및 효율성을 향상시키도록 창안한 프래그먼트 콤비네이션 방법을 제공함에 목적이 있다.

즉, 본 발명의 시스템은 문장의 중심 의미를 추출하는 것을 목적으로 함으로 전체 입력 문장을 커버할 필요가 없이 문장의 핵심 의미를 벗어나지 않는 범위에서 가능한 문장을 길게 커버하는 부분 분석 트리의 조합을 선택함으로써 구문 분석이 실패한 경우에도 의미 추출을 가능하게 하고 휴리스틱(heuristics)을 사용하여 추출 의미의 정확도를 높이며 불필요한 경로의 생성을 차단하여 의미 추출 과정의 효율성을 향상시킨 것을 특징으로 한다.

발명의 구성 및 작용

본 발명은 상기의 목적을 달성하기 위하여 구문 분석이 실패인지 판단하는 단계와, 상기에서 구문 분석이 실패한 경우 휴리스틱에 의거하여 부분 분석 트리로 경로를 형성하는 경로 형성 단계와, 상기에서 생성된 경로중 문장의 중심 의미를 추출할 수 있는 가능한 범위까지 경로를 선택하는 단계와, 상기에서 선택된 경로에 대해 문장의 의미를 추출하는 의미 추출 단계를 수행함을 특징으로 한다.

상기 경로 선택 단계는 부분 분석 트리의 크기가 크고 경로중 미지격의 갯수가 적은 경로를 선택하는 것을 특징으로 한다.

상기 선택된 경로는 문말 용언을 제외한 모든 구를 문말 용언에 접속하는 것을 특징으로 한다.

상기 의미 추출 단계는 동사의 하위 범주 정보와 조사의 격 정보를 이용하여 구가 문장에서 갖는 의미를 추출하는 것을 특징으로 한다.

이하, 본 발명을 도면에 의거 상세히 설명하면 다음과 같다.

도1 은 본 발명의 실시예를 위한 신호 흐름도로서 이에 도시한 바와 같이, 구문 분석이 실패인지 판단하는 단계와, 상기에서 구문 분석이 실패인 경우 부분 분석 트리를 이용하여 경로를 형성하는 경로 형성 단계와, 상기에서 다중 경로의 형성인지를 판단하는 단계와, 상기에서 다중 경로인 경우 문장의 중심 의미를 추출할 수 있는 가능한 범위까지 적절한 경로를 선택하는 단계와, 상기에서 선택된 경로에 대해 문장의 의미를 추출하는 의미 추출 단계를 수행하도록 구성한다.

이와같이 구성한 본 발명의 실시예에 대한 동작 및 작용 효과를 설명하면 다음과 같다.

먼저, 경로 형성(Path Construction) 과정을 설명하면 다음과 같다.

구문 분석기(parser)는 입력문에 대한 구문 분석을 수행하는데, 예를 들어, '철수 삼촌 고향 감'이라는 일정을 나타내는 입력문에 대해 구문 분석 과정을 수행하면 '철수'는 동사 '철수하다'로 분석 가능함으로 형태소 분석기에 의해서 인명을 나타내는 고유명사로 분류되지 않는다.

따라서, 따라서, 구문 분석은 실패하게 되고 문장 전체를 커버하는 분석 트리는 생기지 않는다.

그러나, 구문 분석기(parser)에서 입력문에 대한 구문 분석이 실패한 경우에도 그때까지 진행된 과정이 부분 분석 트리로 남아있게 된다.

즉, '철수 삼촌 고향 감'이라는 일정을 나타내는 입력문에 대해 구문 분석 과정에서 생기는 부분 분석 트리는 도2 와 같다.

따라서, 도2 와 같은 부분 분석 트리를 이용하여 경로를 만든다.

여기서, 경로란 의미 추출을 목적으로 입력 문장에서의 위치를 고려하여 부분 분석 트리를 휴리스틱(Heuristics)을 사용하여 연결한 것을 말하며, 각각의 구문 분석 트리는 구(Phase)라고 부르기로 한다.

도2 에서 'I_1,I_4,I_7,I_8,I_9'는 용언구(VP ; Verbal Phrase)이고 'I_2,I_3,I_5,I_6,I_{10}'은 비용언구(NVP ; Non-verbal Phrase)이다.

특히, 중심 동사가 문말에 위치하는 'I_4,I_7,I_8'은 문말 용언(FVP)이고 'I_1'은 문말용언이 아니다.

이 후, 상기에서 다중 경로가 형성된 경우 경로 선택 과정으로 진행한다.

이때, 경로 선택(Path Selection) 과정에서는 'I_1'부터 'I_{10}'까지의 리스트에서 적절한 조합을 선택하여 문장 전체의 의미를 가장 잘 나타낼 수 있는 분석 트리를 만들어 낸다.

여기서, 부분 분석 트리들이 서로 연결되지 못한 부분이 구문 분석에 실패한 지점을 나타낸다. 즉, 'I_8','I_9','I_{10}' 등이 'I_1'과 서로 연결되지 못한 것은 'I_1'이 고유 명사로 분석되지 못한 것이 원인이다.

따라서, 입력문에 대한 도2 와 같은 예시도에서 가장 길게 문장을 커버하는 조합은 (I_1,I_8),(I_1,I_9),(I_1,I_{10})의 세 개 조합이 가능하다.

상기와 같이 경로를 선택하는 알고리즘은 다음과 같다.

위치 'i'에서 'j'까지의 최대의 경로 'Pi,j'를 구하는 기본 알고리즘은 널리 알려져 있다.

구문 분석시에 부산물로 얻어진 구문 분석 트리들은 구문 분석기의 구현에 따라 달라지지만 본 발명에서는 'I_1,I_2,..,I_{10}'이 리스트 형태로 존재한다고 가정한다.

먼저, 어절 'i'에서 시작하는 모든 구문 분석 트리를 리스트에서 찾는다.

이때, 어절 'i'에서 'j'까지 가는 경로는 어절 'i'에서 직접 도달 가능한 추이적 폐쇄(transitive closure) 각각에서 'j'까지 가는 경로에 어절 'i'와 그 추이적 폐쇄까지의 구문 분석 트리를 합한 것이다.

예를 들어, 도2 의 예시도에서 형태소 Start(=0)에서 End(=4)까지를 연결할 수 있는 최대의 경로 'P0.4'를 찾는 것이 목적으로, 인덱스 '0'으로부터 '4'를 커버하는 가장 긴 경로는 '0'에서 '4'까지 직접 연결하는 링크가 있으면 그 경로이고 그렇지 않으면 어절 '0'에서 직접 연결된 링크와 그곳에서 '4'까지의 경로를 합한 경로가 된다.

가령 'i=0'이라면 도2 의 예시도에서는 'I_1'이 될 것이다.

이를 '0'에서 갈 수 있는 상태(= transitive closure)를 '1'이라고 생각하기로 한다.

이때, 어절 'm=1'에서 'End=4'까지의 경로는 어절 '1'에서 '4'까지 가는 직접 링크가 존재함으로 이 값들로 결정된다.

따라서, 도2 와 같은 예시도에서 어절 '0'에서 '4'까지의 경로는 'I_1'의 구문 분석 트리에 'P1,4'를 합한 것이 된다.

상기와 같이 경로를 선택하는 알고리즘은 도4 의 신호 흐름도와 동일한 과정으로 이루어진다.

즉, 경로 선택 과정이 시작되면 어절 'Start'와 직접 연결된 모든 어절 'm'을 찾고 그 모든 어절 'm'과 'End'사이의 경로를 재귀 호출 방식으로 찾는다.

이에 따라, 어절 'Start'와 'm'을 연결하는 링크와 어절 'm'과 'End'사이의 경로를 합하여 어절 'Start'과 'End'사이의 최대 경로를 구하고 그 구한 경로를 경로 리스트에 더한다.

이 후, 더 처리할 어절 'm'이 있는 경우 상기의 과정을 반복 수행하고 더 처리할 어절 'm'이 없는 경우 이제까지 구한 경로 리스트를 최종 결과로 얻는다.

그러나, 상기와 같은 알고리즘을 적용하는 경우 출력 리스트(list)를 선형 탐색(linear search)함으로 성능의 저하를 초래한다.

이를 개선하기 위해서 도3 의 표와 같은 구조로 어레이를 만들어 참조를 빠르게 한다.

또한, 도4 의 과정과 같이 부분 분석 트리로 경로를 만들고 의미를 추출하는데 있어서 다음과 같은 휴리스틱(Heuristics)을 사용하여 의미를 좀더 정확히 추출한다.

1) 한국어는 SOV 언어로 분류되는 문법 특성상 문장의 주동사가 문말에 위치하는 경향을 가지고 있다.

따라서, 문말 용언이 그 문장의 중심 의미를 표현하고 있다고 볼 수 있다.

2) 한국어에서 수식어는 피수식어의 앞에 위치한다.

즉, 비용언구(NVP)는 뒤에 나오는 용언을 수식하는 것이지 그 앞에 있는 용언구(VP)를 수식하지 않는다.

그러나, 내포문의 경우 비용언구(NVP) 뒤에 나오는 용언이 둘이상이므로 어느 것을 수식하는지 알기 어려우므로 이런 구는 경로를 만들 때 버린다.

따라서, 최종적으로 문말 용언(FVP)에서부터 왼쪽으로 진행하면서 다른 용언이 나오기 전까지의 구를 이용하여 경로를 생성한다.

그러나, 상기에 언급한 사항을 고려해 보면 도 4와 같은 과정을 수행하는 알고리즘은 경로를 모두 형성한 후 불필요한 부분을 제거하는 과정을 수행하여야 한다.

특히, 원래 달랐던 2개의 경로가 불필요한 부분이 제거되어 하나의 동일한 경로가 됨으로 동일한 경로중에서 하나만 남기고 나머지 경로는 없애야 하는 과정을 수행해야 한다.

따라서, 상기의 문제점을 개선하기 위하여 도4 의 알고리즘을 수정하여 도5 와 같은 과정을 수행하는 알고리즘을 제안한다.

즉, 도5 의 알고리즘에서는 모든 경로(Pi.j)만이 아니라 그 경로내에 포함된 동사의 갯수(numVerb)도 궤환받는다.

이에 따라, 동사의 갯수(numVerb)를 참조함으로써 불필요한 부분을 경로에 포함시킴이 없이 필요한 부분만을 생성하며 중복되는 동일 경로를 제거하는 과정도 없앨 수 있다.

또한, 도5 와 같은 알고리즘은 현재의 부분 분석 트리가 용언이면 경로내에 포함된 용언이 없을 때에만 용언을 더함으로 경로에는 항상 마지막 용언과 또 하나의 용언이 나오기 전까지의 구문 분석 트리만이 더해지게 된다.

따라서, 도2 와 같은 예시도에서 (l_1)과 (l_8),(l_9)의 3개의 경로가 선택된다.

상기에서 'l_{10}'이 선택되지 않는 것은 'l_{10}'이 용언구가 아님으로 최초 나오는 용언구 'l_1'을 만났을 때 만들어 오던 모든 경로를 버리고 'l_1'을 문말 용언으로 결정했기 때문이다.

상기와 같은 과정으로 생성된 모든 경로에서 의미를 추출하면 애매성이 증가함으로 입력문의 중심 의미를 추출하기 위하여 아래와 같은 2가지의 휴리스틱을 사용하여 생성된 경로중 일부만을 선택한다.

(1) 실제 경로를 구성하는 길이에 대한 부분 분석 트리의 갯수가 가장 적은 것을 선택한다.

(2) 경로내에 포함된 구중에서 구의 격을 알 수 없는 것의 갯수가 적은 것을 선택한다.

이는 구문 분석에 실패한 문장이지만 성공한 경우에 최대한 근사시키기 위한 휴리스틱으로 볼 수 있다.

상기에서 (1)의 의미는 입력문을 최대한 길게 포함하여 분석한 것을 선호한다는 것이고 (2)의 의미는 문장내에서 역할을 알기 힘든 구가 적을수록 좋다는 휴리스틱을 반영하고 있다.

따라서, 도2 의 예시도에서 만들어진 3개의 경로(l_1),(l_8),(l_9)중에서 전체 길이에 비해 부분 분석 트리의 갯수가 적은 (l_8)과 (l_9)의 2개의 경로가 최종 경로로 선택된다.

즉, 상기 도5의 알고리즘에 의한 과정을 간략히 설명하면 다음과 같다.

먼저, 경로 선택 과정이 시작되면 어절 'Start'와 직접 연결된 모든 어절 'm'을 찾고 그 모든 어절 'm'과 'End'사이의 경로를 재귀 호출 방식으로 찾고 그 경로내의 용언수(numVerb)를 찾는다.

이때, 어절 'Start'과 'm'을 연결하는 링크가 용언인가를 판단한다.

만일, 어절 'Start'과 'm'을 연결하는 링크가 용언인 경우 용언수(numVerb)가 '0'인지 판별한다.

이에 따라, 용언수(numVerb)가 '0'이면 어절 'Start'와 'm'을 연결하는 링크를 경로 리스트에 더한 후 용언수(numVerb)를 '1'증가시킨 후 더 처리할 어절 'm'이 있는지 판별한다.

상기에서 용언수(numVerb)가 '0'인 경우 용언수(numVerb)를 '1' 증가시킨 후 더 처리할 어절 'm'이 있는지 판별한다.

그리고, 상기에서 어절 'Start'과 'm'을 연결하는 링크가 용언이 아닌 경우 용언수(numVerb)가 '2'보다 작은지 판별한다.

이에 따라, 용언수(numVerb)가 '2'보다 크면 더 처리할 어절 'm'이 있는지 판별하고 '2'보다 작으면 어절 'Start'와 'm'을 연결하는 링크와 어절 'm'과 'End'사이의 경로를 합하여 어절 'Start'과 'End'사이의 최대 경로를 구한 수 그 구한 경로를 경로 리스트에 더한다.

이 후, 더 처리할 어절 'm'이 있는 경우 상기의 과정을 반복 수행하고 더 처리할 어절 'm'이 없는 경우 이제까지 구한 경로 리스트를 최종 결과로 얻는다.

이 후, 상기와 같은 과정으로 최종적인 경로(l_8)과 (l_9)가 선택되면 의미 추출 과정으로 진행한다.

구(Phrase)는 용언구(VP ; Verbal Phrase)와 비용언구(NVP ; Non_verbal Phrase)로 구분된다.

특히, 경로내에 용언구는 최대 1개가 존재할 수 있는데 이는 문말용언(FVP)이다.

그리고, 비용언구(NVP)는 명사구(NP), 부사구(PP), 명사(NOUN) 그리고 기타의 구로 구분하는데, 이는 그것으로부터 얻을 수 있는 정보가 서로 다르기 때문이다.

상기에서 명사구(NP)는 '명사 + 주격조사/목적격 조사'의 결합형태, 부사구(PP)는 '명사 + 부사격 조사'의 결합 형태, 명사(NOUN)는 조사없이 명사 단독으로 쓰였거나 '명사 + 보조사'의 결합 형태이다.

따라서, 상기에서 최종 선택된 경로에 용언구가 존재하는 경우 그 용언구로부터 하위 범주화 정보를 얻을 수 있다.

하위 범주화 정보는 문법적/의미적인 격으로 어떤 것을 필요로 하는가를 나타낸다.

예를 들어 동사 '가다'의 하위 범주 정보는 아래와 같으며 이는 사전에 등록되어 있다.

SUBJ{agt : 사람}PP{loc : 장소}이는 주격으로 사람을 나타내는 말이 와야 하고 그때의 문장에서 의미적 역할인 경우(case)가 'agent'이며, 부사격으로는 장소를 나타내는 말이 와야 하고 그때의 경우(case)가 'location'이 된다는 것을 의미한다.

그리고, 명사구에서 격조사에 따라 주격 조사 '이/가'가 쓰였으면 주격(SUBJ), 목적격조사 '을/를'이 쓰였으면 목적격(OBJ)으로 정할 수 있다.

즉, 조사에 따라 문법적 기능이 무엇인지를 알 수 있다.

또한, 부사구에서는 문법적 기능이 부사격이며 부사격 조사가 가지는 의미에 따라 의미적인 격을 알 수 있다.

이때, 많은 경우 하나의 부사격 조사가 나타내는 의미격은 복수개일 때가 많다.

예를 들어, 부사격 조사 '에'는 'time, scale, target, location, reason, manner'등 다양한 격을 나타낼 수 있다.

그 외에도 부사어중에서 '3월 8일', '지난달', '어제'등과 같이 시간을 나타내는 부사어는 전처리를 통해서 시간구임을 알 수 있는데, 이런 부류는 경우(case)가 'time'을 나타내는 것으로 보고 부사구(PP)와 동일시하여 처리한다.

그러나, 명사(NOUN)나 격을 알 수 없는 기타에 속하는 것들은 그 문법적 기능이 무엇인지 알 수 없다.

한편, 하나의 경로로부터 의미를 추출할 때 용언이 요구하는 의미적 역할을 나타내는 경우(case)중에서 채워진 것과

그렇지 않은 것을 나타내는 filledCase/unfilled Case 와 문법적 기능을 나타내는 filledGF/unfilledGF 의 리스트를 이용하여 관리한다.

이는 용언이 요구하는 case 정보중에서 이미 채워진 것과 아직 채워지지 않은 것들을 나타내며 GF(Grammatical Function)는 하위 범주가 요구되는 문법 기능이 {주격(SUBJ), 목적격(OBJ), 부사격(PP)}중 어느 것인가를 나타낸다.

따라서, 상기에서 선택된 각각의 경로(l_8,l_9)에 대해 다음과 같은 과정을 통해 구를 처리하여 구가 문장에서 갖는 의미를 추출한다.

(1) 경로내에 문말 용언이 존재하지 않으면 더미(dummy)의 용언구(VP)를 경로의 맨 끝에 하나 추가하고 filledCase와 filledGF 에는 'NULL'을 세팅하며 unfilledCase 와 unfilledGF 에는 'ALL'을 세팅한다.

(2) 용언구(VP)를 기준으로 가까운 것부터 먼 비용언구(NVP)의 순으로 처리하는데, 이는 용언과 가까이 쓰인 구가 그 용언을 수식할 확률이 높다고 가정하기 때문이다.

(3) 비용언구(NVP)는 명사구(NP)부터 먼저 처리하는데, 이것은 부사격(PP)보다 주격(SUBJ)이나 목적격(OBJ)의 사격성이 더 높다고 판단되기 때문이다.

이때, 용언구(VP)가 요구하는 하위 범주(subcategory)의 case 정보와 명사구(NP)에서 조사가 제공하는 case 정보가 매치되는 격으로 결정한다.

만일, 용언구(VP)가 요구하는 정보와 매치되지 않거나 용언구(VP)가 없으면 명사구(NP)의 조사에 의해서 얻을 수 있는 격으로 결정하고 조사가 없으면 미지격(unknown)으로 처리한다.

(4) 부사구(PP)를 처리한다.

(5) 나머지 구에 대해서 처리한다.

상기에서 명확한 의미 추출을 위하여 다음과 같은 휴리스틱을 이용한다.

1) 상기의 과정을 통해서 비용언구(NVP)의 정보와 용언구(VP)에서 요구하는 정보가 서로 같으면 문법 기능이나 의미격이 정해지게 된다.

이때, filledGF/filledCase 에 그 정보를 추가하고 unfilledGF/unfilledCase 에서 해당 엔트리를 삭제한다.

만일, 매치되는 정보가 없으면 그 구를 미지격(unknown)으로 설정한다.

2) 부사격 조사와 같이 복수개의 격을 갖는 경우에는 다른 구에 의해서 이미 채워진 의미격이 있으면 그 격을 제외한 의미격으로만 정한다.

발명의 효과

상기에서 상세히 설명한 바와 같이 본 발명은 구문해석과정이 실패로 끝났을 때에도 문말 용언을 중심으로 휴리스틱에 의거하여 가능성이 있는 경로만을 생성한 후 이 중에서 선택된 경로만으로 문장의 중심 의미 추출에 사용함으로써 성능을 향상시킬 수 있는 효과가 있다.

즉, 본 발명은 휴리스틱에 의거하여 가능성이 없는 불필요하다고 생각되는 경로는 가지 치기(pruning)를 하여 생성하지 않고 가능성이 있는 경로만을 생성한 후 부분 분석 트리의 크기가 크고 경로중에서 미지격의 갯수가 적은 경로만을 선택함으로써 구문 분석기의 중간 과정에서 남은 정보를 최대한 활용하며 또한, 선택된 경로는 문말 용언을 제외한 모든 구를 문말 용언에 접속시킨 후 동사의 하위 범주 정보와 조사의 격 정보를 이용하여 구가 문장에서 갖는 의미를 추출함으로써 구문 분석의 성능을 향상시킬 수 있다.

(57)청구의 범위

청구항1

구문 분석이 실패인지 판별하는 단계와, 상기에서 구문 분석이 실패로 판별한 경우 부분 분석 트리를 이용하여 경로를 형성하는 경로 형성 단계와, 상기에서 생성된 경로에서 문장의 의미를 추출할 수 있는 가능한 경로만을 선택하는

단계와, 상기에서 선택된 경로에서 문장의 중심 의미를 추출하는 의미 추출 단계를 수행함을 특징으로 하는 프래그먼트 콤비네이션 방법.

청구항2

제1항에 있어서, 경로 선택 단계는 입력문을 최대한 길게 포함하도록 부분 분석 트리의 크기가 크고 경로에 포함된 구중 미지격의 갯수가 적은 경로를 선택하는 것을 특징으로 하는 프래그먼트 콤비네이션 방법.

청구항3

제1항에 있어서, 경로 선택 단계는 경로내에 포함된 동사의 갯수를 참조하여 문말 용언을 제외한 모든 구를 문말 용언에 접속함에 의해 불필요한 경로를 제거하는 것을 특징으로 하는 프래그먼트 콤비네이션 방법.

청구항4

제1항 또는 제3항에 있어서, 경로는 현재의 부분 분석 트리가 용언이면 하나의 용언이 나오기 전까지의 부분 분석 트리를 더하여 선택하는 것을 특징으로 하는 프래그먼트 콤비네이션 방법.

청구항5

제1항에 있어서, 의미 추출 단계는 동사의 하위 범주 정보와 조사의 격 정보를 이용하여 구가 문장에서 갖는 의미를 추출하는 것을 특징으로 하는 프래그먼트 콤비네이션 방법.

청구항6

제5항에 있어서, 동사의 하위 범주 정보는 문법적/의미적인 격으로 어떤 것을 필요로 하는가를 나타내는 것을 특징으로 하는 프래그먼트 콤비네이션 방법.

청구항7

제1항 또는 제5항에 있어서, 의미 추출 단계는 아래의 순서로 수행하는 것을 특징으로 하는 프래그먼트 콤비네이션 방법.

(1) 경로내에 문말 용언이 존재하지 않으면 더미(dummy)의 용언구(VP)를 경로의 맨 끝에 하나 추가하고 filledCase와 filledGF 에는 'NULL'을 세팅하며 unfilledCase 와 unfilledGF 에는 'ALL'을 세팅한다.

(2) 용언구(VP)를 기준으로 가까운 것부터 먼 비용언구(NVP)의 순으로 처리한다.

(3) 비용언구(NVP)는 명사구(NP)부터 먼저 처리한다.

(4) 부사구(PP)를 처리한다.

(5) 나머지 구에 대해서 처리한다.

청구항8

제7항에 있어서, 다음과 같은 휴리스틱을 이용하여 명확한 의미 추출을 수행함을 특징으로 하는 프래그먼트 콤비네이션 방법.

1) 비용언구(NVP)의 정보와 용언구(VP)에서 요구하는 정보가 서로 같으면 정해진 문법 기능이나 의미격의 정보를 filledGF/filledCase 에 추가하고 unfilledGF/ unfilledCase 에서 해당 엔트리를 삭제한다.

만일, 매치되는 정보가 없으면 그 구를 미지격(unknown)으로 설정한다.

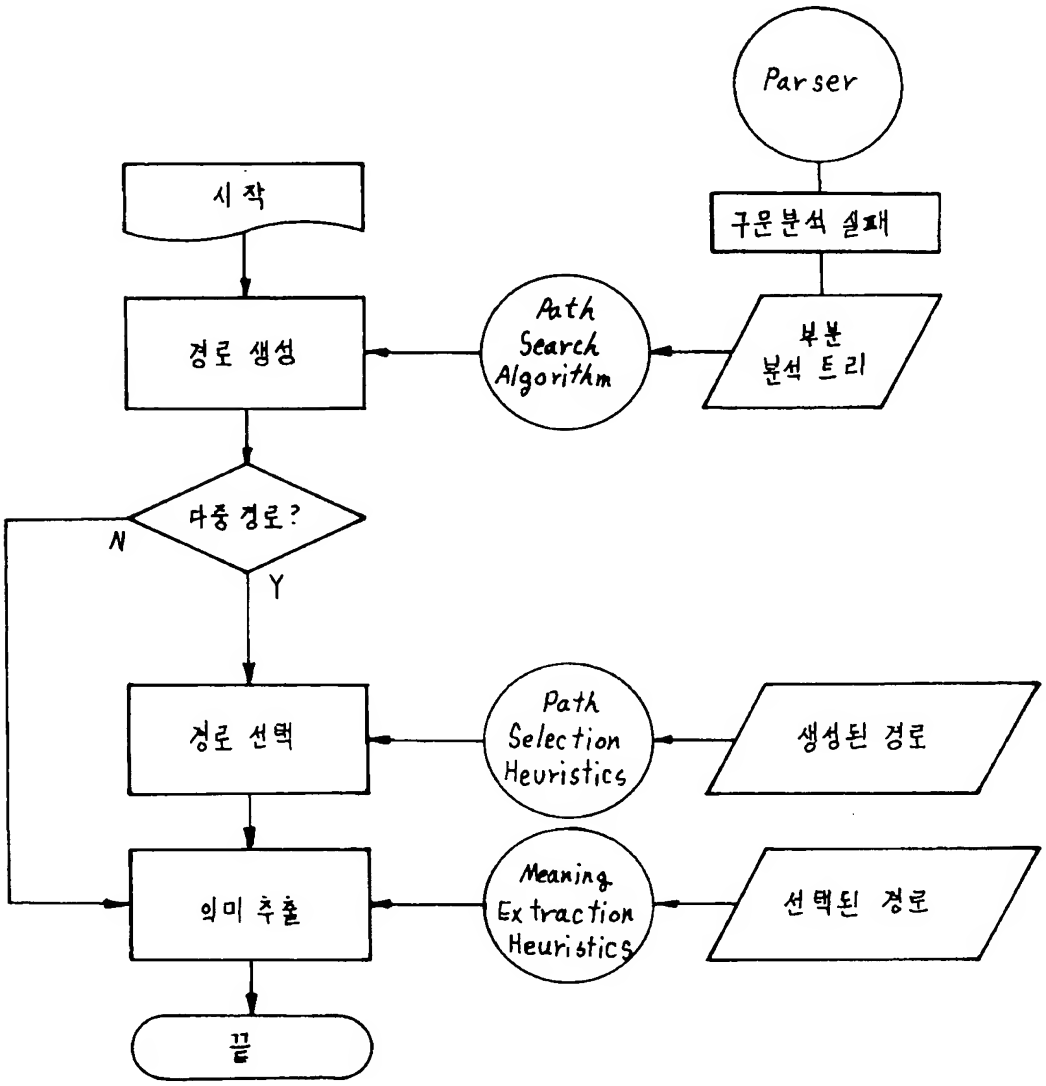
2) 부사격 조사와 같이 복수개의 격을 갖는 경우 다른 구에 의해서 이미 채워진 의미격이 있으면 그 격을 제외한 의미격으로만 정한다.

청구항9

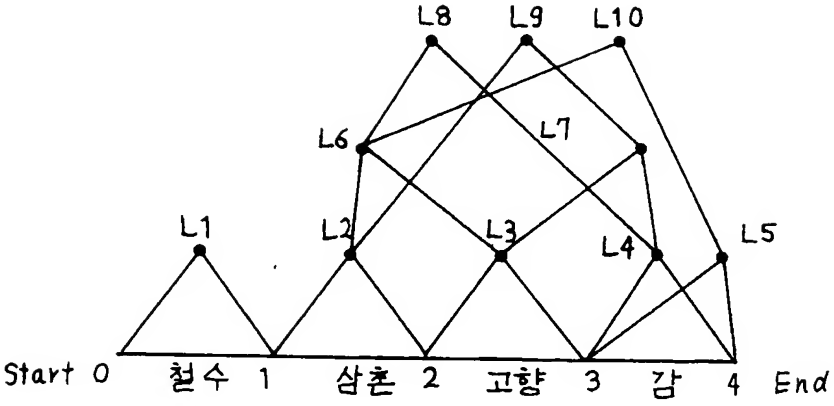
구문 분석이 실패인지 판별하는 단계와, 상기에서 구문 분석이 실패한 경우로 판별한 경우 부분 분석 트리를 이용하여 경로를 형성하는 경로 형성 단계와, 상기에서 생성된 경로가 다중 경로인지 판단하는 단계와, 상기에서 다중 경로로 판별한 경우 문장의 의미를 추출할 수 있는 가능한 경로만을 선택하는 단계와, 상기에서 단일 경로 또는 상기에서 선택된 경로에서 문장의 중심 의미를 추출하는 의미 추출 단계를 수행함을 특징으로 하는 프래그먼트 콤비네이션 방법.

도면

도면1



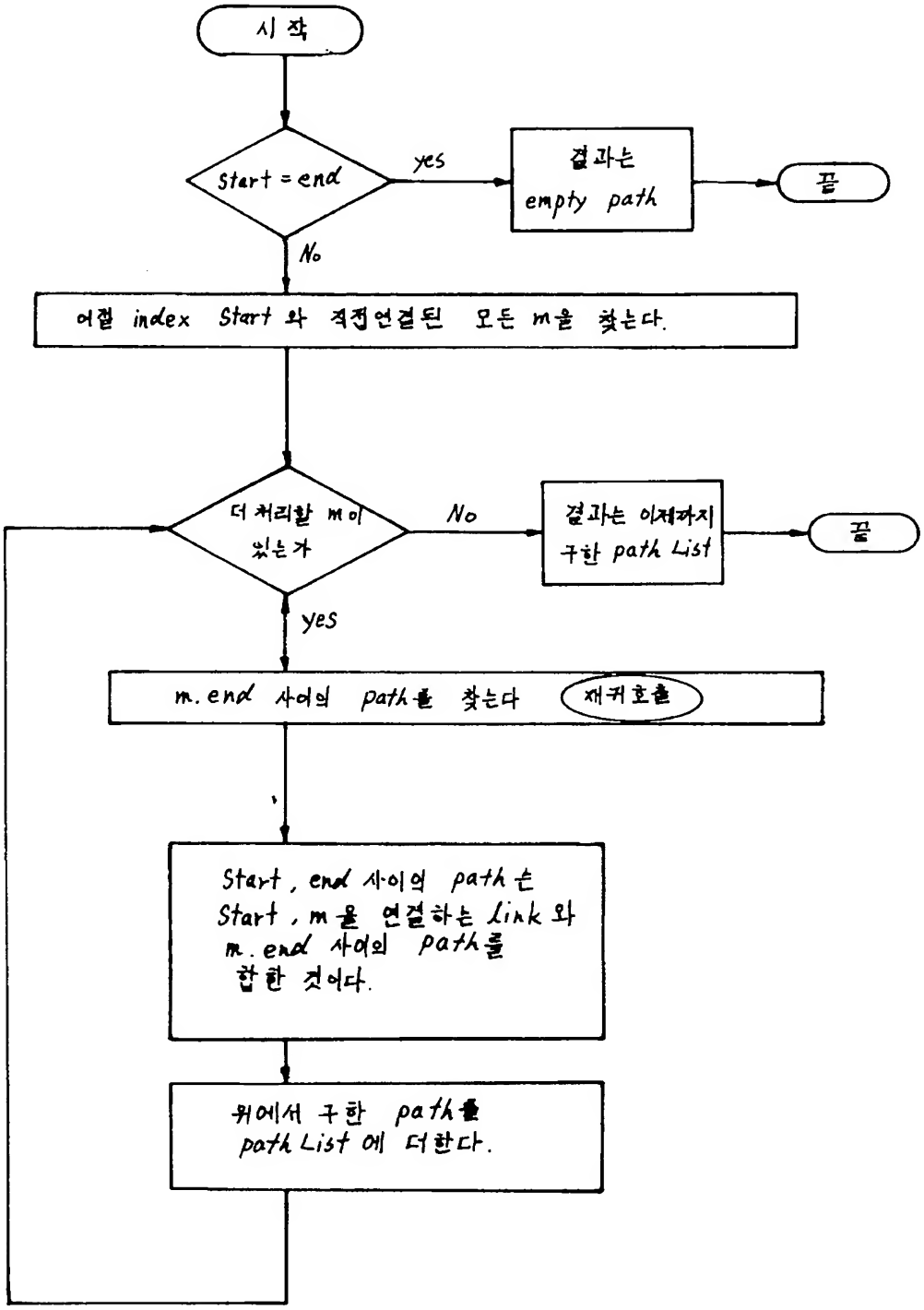
도면2



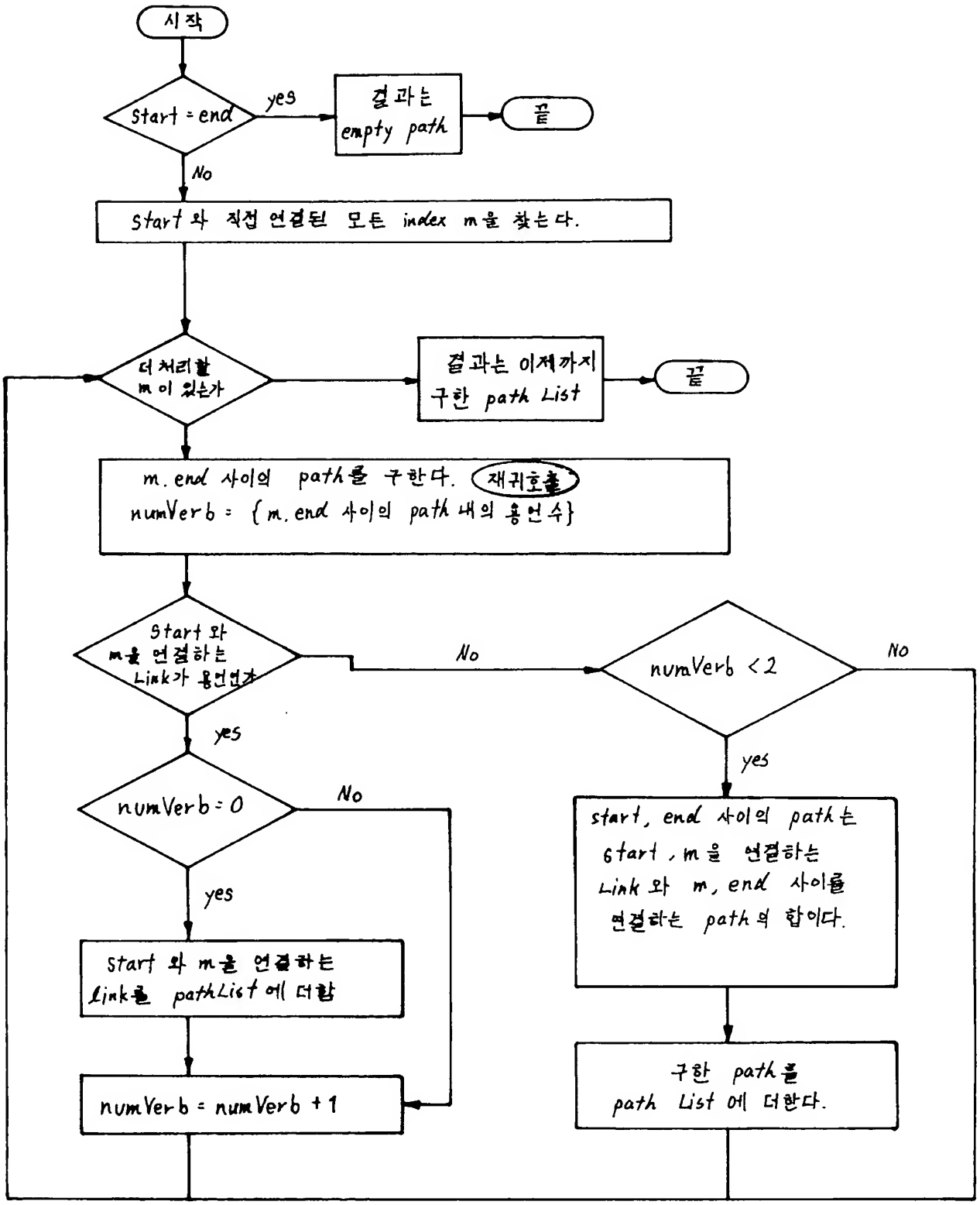
도면3

index	transitive closure	Link
0	1	l_1
1	{4, 4, 4}	$\{l_8, l_9, l_{10}\}$
4	END	-

도면4


시작
Start = end
yes
결과는 empty path
끝
No
어떤 index Start 와 직접연결된 모든 m을 찾는다.
더 처리할 m이 있는가
No
결과는 이제까지 구한 path List
끝
yes
m.end 사이의 path를 찾는다
재귀호출
Start, end 사이의 path는 Start, m 을 연결하는 link 와 m.end 사이의 path를 합한 것이다.
위에서 구한 path를 path List 에 더한다.

도면5


시작
Start = end
yes
결과는 empty path
끝
No
Start 와 직접 연결된 모든 index m을 찾는다.
더 처리할 m이 있는가
결과는 이제까지 구한 path List
끝
m, end 사이의 path를 구한다. 재귀호출
numVerb = {m, end 사이의 path 내의 용언수}
Start 와 m을 연결하는 Link가 용언인가
No
numVerb < 2
No
yes
yes
numVerb = 0
No
start, end 사이의 path는 Start, m을 연결하는 Link와 m, end 사이를 연결하는 path의 합이다.
yes
Start 와 m을 연결하는 Link를 pathList에 더함
구한 path를 path List에 더한다.
numVerb = numVerb + 1